

## PATENT ABSTRACTS OF JAPAN

③

(11)Publication number : 2002-124996

(43)Date of publication of application : 26.04.2002

(51)Int.Cl.

H04L 12/66

G06F 13/00

(21)Application number : 2000-350265

(71)Applicant : BABA YOSHIMI

(22)Date of filing : 13.10.2000

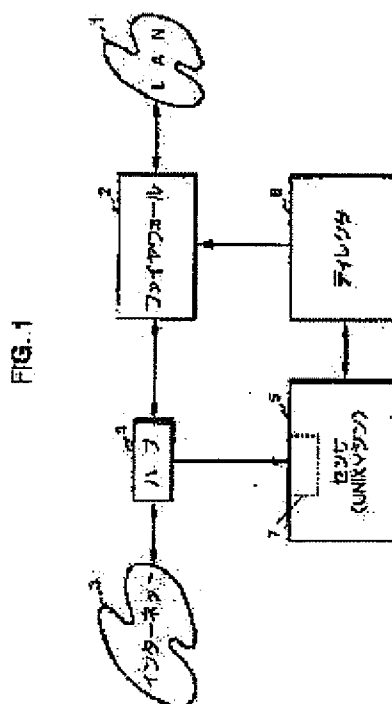
(72)Inventor : BABA YOSHIMI

## (54) FAST PACKET ACQUIRING ENGINE/SECURITY

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a cracker monitor system of a simple system configuration to protect an LAN 1 from attacking by a cracker by automatically detecting the attack by the cracker to the LAN 1 with no burdensome limit on communication or experienced engineers.

**SOLUTION:** A sensor 5 is provided where a hash algorithm is used to sequentially acquire IP packets passing an entrance of a LAN 1. The sensor 5 quickly detects various attacks by a cracker to the LAN 1 based on the acquired IP packet. The information related to the attack which is detected by the sensor 5 is provided to a director 6 controlling a fire wall 2. The director 6 controls setting of the fire wall 2 according to the supplied information and prevents an IP packet related to the detected attack from entering the LAN 1.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2002-124996  
(P2002-124996A)

(43) 公開日 平成14年4月26日 (2002. 4. 26)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
H 0 4 L 12/66		H 0 4 L 12/66	B 5 B 0 8 9
G 0 6 F 13/00	3 5 1	G 0 6 F 13/00	3 5 1 Z 5 K 0 3 0

審査請求 未請求 請求項の数6 書面 (全 14 頁)

(21) 出願番号 特願2000-350265 (P2000-350265)  
(22) 出願日 平成12年10月13日 (2000. 10. 13)

(71) 出願人 599098415  
馬場 芳美  
横浜市港北区太尾町644  
(72) 発明者 馬場 芳美  
神奈川県横浜市港北区太尾町644  
F ターム (参考) 5B089 GA04 HA04 HA10 HB02 KC18  
KH28 MC08  
5K030 GA14 HA08 HC01 HC14 HD03  
HD06 JA10 KA06 KA13 LC14  
LC15 LC16 MA04 MB18

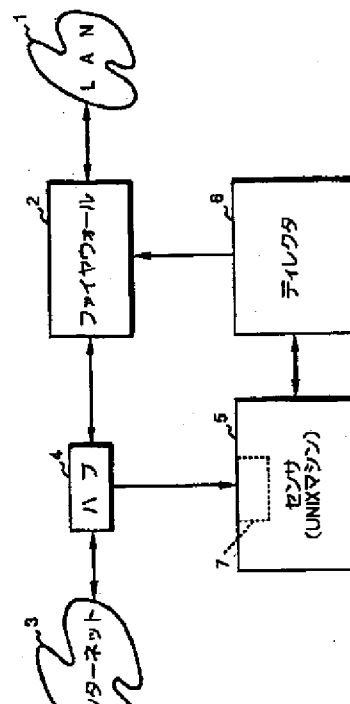
(54) 【発明の名称】 高速パケット取得エンジン・セキュリティ

(57) 【要約】

【課題】 LAN 1 に対するクラッカーからの攻撃を自動的に検知し、通信を必要以上に制限したり、熟練技術者による労力を必要とすることなく、簡易なシステム構成でクラッカーからの攻撃に対する LAN 1 の保護を図ることができるクラッカー監視システムである。

【解決手段】 ハッシュアルゴリズムを用いる事で、LAN 1 の入り口にそこを通る IP パケットを逐次取得するセンサ 5 を設ける。センサ 5 は、取得した IP パケットに基づき、LAN 1 に対するクラッカーからの各種攻撃を高速に検知する。センサ 5 が検知した攻撃に関する情報は、ファイアウォール 2 を制御するディレクタ 6 に与えられる。ディレクタ 6 は与えられた情報に応じてファイアウォール 2 の設定を制御し、検知された攻撃に係る IP パケットが LAN 1 に進入するのを阻止する。

FIG. 1



## 【特許請求の範囲】

【請求項1】ネットワークを高速に流れるパケットを通過させる機器に於いて、そのパケットを遅滞なく通過させる時間内に、それを検出分類し、適切な処理を施す事が出来る事に特徴を有する、通信用装置。

【請求項2】送信および受信情報をハッシュ法によって圧縮し、充分狭いメモリー空間内に全情報を展開する事を可能とした事に特徴を有する、上記、請求項1に記載の通信用装置。

【請求項3】ハッシュ表作製時に、2重ハッシュとリスト方式を利用する事により、高効率を達成する事に特徴を有する、上記、請求項1及び／乃至は2に記載の通信用装置。

【請求項4】ハッシュ表利用時に圧縮率を制御し、メモリー利用率を約80%に保つ事で、衝突を避けつつ、高効率を達成する事に特徴を有する、上記、請求項1及び／乃至は2及び／乃至は3に記載の通信用装置。

【請求項5】インターネット上のハッキング乃至はクラッキングと言われる攻撃、特に、TCP-Syn Flood、Teardrop、Land、Ping of Death、Distributed Denial of Serviceを、検出及び／乃至は遮断する事を目的とした上記、請求項1及び／乃至は2及び／乃至は3及び／乃至は4に記載の通信用安全装置。

【請求項6】インターネット上の通信を利用して、OSのバグ等のバッファオーバーフローを起こさせるなどして、ルートのパスワードを奪取するなどの攻撃が起こった時に、高速にそれに関する通信データを補足する事に特徴を有する上記、請求項1及び／乃至は2及び／乃至は3及び／乃至は4及び／乃至は5に記載の通信用安全装置。

## 【発明の詳細な説明】

## 【001】

【発明の属する技術分野】 本発明は、クラッカーによるインターネットを介したネットワーク（LAN）への攻撃を監視し、さらにはその攻撃からネットワークを保護するためのシステムに関する。

【従来の技術】 近年、企業などの組織内に構築されたネットワーク（LAN）は、その多くがインターネットに接続され、他のネットワーク等との間での各種情報のやりとり（通信）がインターネットを介して行われている。この通信では、一般に、所謂OSI階層モデルにおけるネットワーク層に主として対応するプロトコルとしてIP（Internet Protocol）が用いられ、通信データはIPパケットの形態でやりとりされる。そして、上記ネットワーク層の上位のトランスポート層に主として対応するプロトコル（IPの上位のプロトコル）として、TCP（Transmission Control Protocol）あるいはUDP（User Datagram Protocol）を用いるのが通例である。

用いるのが通例である。

【002】この種のネットワークは、インターネット上のサーバや他のネットワークなどとの間で、多種多様な情報のやりとりを低コストで行うことができるという利点を有する。反面、インターネットが極めて高度な公開性を有することから、所謂クラッカーからの攻撃を受ける危険性にさらされることとなる。このため、そのような攻撃からネットワークを保護することが要求される。このようなネットワークの保護を行うためのシステムとしては、従来、保護しようとするネットワークの入り口に、ファイアウォール（詳しくはファイアウォールの機能をもたせたコンピュータ）を設けたシステムが知られている。このファイアウォールは、あらかじめネットワーク管理者などが定めた種類の通信がネットワーク内とその外部との間で行われるのを阻止し、それ以外の許可された通信のみをネットワーク内とその外部との間で行うことができるようにするものである。この場合、阻止する通信の種類は、例えばIPパケットに含まれる送信元IPアドレスや宛先IPアドレス、宛先ポート番号などによって指定可能とされている。このようなファイアウォールによれば、ネットワーク内の特定のIPアドレスを有するホスト（コンピュータ）、あるいはそのホストの特定のポート番号に対する外部からのアクセスを禁止したり、ネットワークの外部の特定のIPアドレス以外のIPアドレスからのネットワークへのアクセスを禁止したりすることができる。従って、ネットワークへの進入を禁止する通信データの種類をファイアウォールに対して適切に設定しておけば、ネットワークへの攻撃の危険性を低減することが可能である。

【003】更にこのようなネットワークへの攻撃を検出するためのシステムとしては、従来、保護しようとするネットワークの入り口に、侵入検知システム（英語ではintrusion detection system、詳しくは侵入者の通信パターンを検出する機能をもたせたコンピュータ）を設けたシステムが知られている。この侵入検知システムは、あらかじめ収集された種類の攻撃者に特有のパターンの通信がネットワーク内とその外部との間で行われるのを検出し、それを管理者に通報するものである。ここで、その検出には、データの収集およびデータベースの参照等の時間を要する為、攻撃が行われた事の検出に基づいてそれを遮断する、或いは、それ以外の許可された通信のみをネットワーク内とその外部との間で行うことができるようにするのは、通常不可能である。この場合、通信を阻止する為には、例えばIPパケットが通過するまでのかなり短時間の間に、通信に含まれる情報などによって検出特定が可能とされなくてはならない為、通常のパケット確認用のツールであるスニフアー（sniffer）あるいはBPF（Berkeley Packet Filter）あるいは

侵入検知システムによっても、ネットワーク内とネットワークの外部で、ネットワークへの侵入を禁止する事は、適切に設定しておいても、ネットワークへの攻撃の危険性を減らすことはできても、なくすことは不可能である。つまり、ファイアウォールや侵入検知システムにより防御するには、保護しようとするネットワーク内の各ホストがどのような情報を利用し、もしくは外部に提供し、また、ネットワーク内のどのような情報を保護すべきか、予想される攻撃としてどのようなものが想定されるか、ということなどを総合的に考慮して決定しなければならないし、かなりの熟練技術者によっても場合によっては不可能な事情があった。

【004】従って、ネットワークの管理運営には、常時、攻撃される事を前提とした修復を伴う、熟練技術者による多大な労力やコストを要するものとなっていた。また、上記のような従来のファイアウォールは、攻撃の可能性のある通信をすべて排除しようとするものである。従って、設定により禁止された種類の通信は、その通信がクラッカーからの攻撃によるものであるか否かにかかわらず一律的に排除される。つまり、ネットワークと外部との通信の自由度が必要以上に制限される。このため、ファイアウォールを備えたネットワークでは、インターネット上の利用可能な情報提供サービスの制限を受ける。この結果、インターネット上の多くの情報資源を有用に享受することができないという不都合を生じるものであった。

#### 【005】

【発明が解決しようとする課題】 本発明はかかる背景に鑑みてなされたものであり、ネットワークに対するクラッカーからの攻撃を自動的に検知し、通信を必要以上に制限したり、熟練技術者による労力を必要とすることなく、簡易なシステム構成でクラッカーからの攻撃に対するネットワークの保護を図ることができるクラッカー監視攻撃遮断システムを提供することを目的とする。

【課題を解決するための手段】 本発明のクラッカー監視システムは、かかる目的を達成するために、IP (Internet Protocol) に基づく通信を行うネットワークの入り口において該入り口を通過するIPパケットを逐次取得して累積的に保持し、保持した複数のIPパケットを監視することにより該ネットワークに対するクラッカーからの攻撃を検知する攻撃検知手段と、該攻撃検知手段が前記攻撃を検知したとき、それに応じた所定の処理を行う処理手段とを備えたことを特徴とするものである。

【006】すなわち、本願発明者等がクラッカーによる各種攻撃の手法を検討したところ、一般に、多くの種類の攻撃は、それぞれその攻撃の際に時系列的に通信される複数のIPパケットに特徴的な相互関連性を有する。

累積的に保持し、その保持した複数のIPパケットを監視することで、クラッカーによる前記ネットワークへの攻撃をリアルタイムで検知することができる。そして、このように攻撃を検知できれば、それに応じて前記処理手段により適当な処理（例えばネットワーク管理者などへの報知や、クラッカーによる通信を遮断する処理等）を行うことで、その攻撃からのネットワークの保護を図ることができる。この場合、クラッカーによる攻撃を十分精度よく検知防御する為には、一般にかなりの高速度を要するに進行する。このため、攻撃を検知するために、高速度にIPパケットに関する情報を蓄積して行くテクニックとして、ハッシュ表のアルゴリズムを要する。あるいは、それによってネットワークを保護するための処置を行えば、ネットワークの損害を十分に抑えることができる。

【007】このような本発明のシステムによれば、クラッカーによる攻撃をリアルタイムで検知できるので、その検知がなされたとき、且つそのときのみ攻撃に対する対策処置を施せばよい。このため、ネットワーク管理者等は、所謂ログファイル（通信記録簿）等を頻繁に参照したりする必要性が低減される。さらに、ネットワークの構築や再編等の際に、クラッカーによる攻撃を予測的に考慮するような労力が軽減される。また、攻撃が検知されない通常時は、ネットワークとその外部との通信を、攻撃の可能性を予測して制限する必要性がなく、その通信の自由度を高めることができる。従って、本発明によれば、ネットワークに対するクラッカーからの攻撃を自動的に検知し、通信を必要以上に制限したり、熟練技術者による労力を必要とすることなく、簡易なシステム構成でクラッカーからの攻撃に対するネットワークの保護を図ることができる。かかる本発明においては、前記攻撃検知手段は、前記ネットワークの入り口を通過する全てのIPパケットを受信可能に構成しておく。これにより、クラッカーによる多くの種類の攻撃を速やかに検知することが可能となる。さらに、本発明では、前記攻撃検知手段は、IPパケットの受信のみが可能に構成しておく。

【008】これによれば、前記攻撃検知手段は、自己のIPアドレスやMAC (Media Access Control) アドレス等、自己情報のデータをネットワークに送信することがないため、クラッカーなどによりその存在が認識されたり、攻撃の対象とされることがない。従って、攻撃検知手段の安全性を確保し、ひいては、本発明のシステムの信頼性を確保することができる。また、本発明では、前記攻撃検知手段は、複数の種類の前記攻撃に対して、各種の攻撃を検知するためのアルゴリズムを保持しており、取得して保持した前記複数のIPパケットから前記アルゴリズムに基づき各種の攻撃を検知する。これにより、クラッカーによる前記

ワークの安全性を高めることができる。また、前記アルゴリズムを適宜更新することで、新しい種類の攻撃に対しても対応することが可能となる。この場合、前記攻撃検知手段は、取得して保持した複数のIPパケットを少なくとも送信元IPアドレス及び／又は宛先IPアドレスにより分類する手段として、二重型リスト・ハッシュ方式を具備し、その分類した複数のIPパケットの為に表から前記各種類の攻撃を検知する。

【009】すなわち、複数の種類の攻撃を検知するためには、IPパケットの送信元IPアドレスや宛先IPアドレス（これらはIPパケットのIPヘッダに付与されている）が重要な鍵となることが多い。従って、所定時間内に取得したIPパケットを送信元IPアドレス及び／又は宛先IPアドレスにより分類して保持することで、それらのIPパケットから攻撃を検知しやすくなる。本発明では、より具体的には、前記攻撃検知手段は、次のようにハッシュ表によって攻撃を検知する。ハッシュ法（hashing）は、メモリ上でデータを高速に検索するための手法である。各種のツリー構造とは異なり、静的な配列だけで簡単に実装でき、効率も極めて高い。配列上でデータを検索する手法は、いくつかある。以下、単純な手法から順に説明していき、ハッシュ法の説明に至る。ある情報をメモリ上で処理する場合を考えてみよう。情報のキーとして番号（整数）を用いることだけを決めておき、その他については考えないことにする。

#### （1）単純配列

データの出現順に配列につめていく。もっとも単純かつ基本的な方法。データの挿入は高速だが、検索は端から順に見ていく（リニアサーチという）しかないため、平均するとデータ件数の半分について処理が必要となる。この手法は遅いので、データ件数が多い場合には使われない・・・と良いのだが、多くのプログラマがこの方法しか知らないが故に、現実には件数が多い場合にも使われている。

#### （2）ソート済み配列

あらかじめ配列上のデータをキー（この場合は社員番号）の順に整列しておく。こうすると、データの挿入には時間がかかる（後述）が、検索にバイナリサーチという手法が使えるので、高々 $\log(N)$ 回の処理ですむ。 $100$ 万件のデータでも $\log(N) \approx 20$ だから膨大なデータでも非常に高速である。一方、データの維持にはコスト（処理時間）がかかる。データの内容が固定的なもの（例：Visual Basicのキーワード…printなど）である場合や、データが発生するフェーズと参照されるフェーズがはっきり分れている場合（DXFの線種や複合図形定義）には、データをまとめてソートできるから、クイックソートなどの高速アルゴリズムを用いれば $N \cdot \log(N)$ の手間で済む。

ためつつ使用する場合、ソート済みの配列にデータを挿入するしかなく、処理時間は $N$ の二乗のオーダーを要する、つまり遅い。

#### （3）逆引き表

小さなデータという前提なので、特殊なケースとして、番号が3桁の整数である場合を考えてみよう。この場合、番号は $001 \sim 999$ の $1000$ 種類しかない。このため、あらかじめ $1000$ 要素の配列を用意しておき、番号をインデックスとして配列に入れてしまうという方法がある。これを逆引き表という。情報をいれる場合の擬似コードは以下になる。

マスター[番号] = 内容

逆引き表の利点は、登録も検索も極めて高速であり、処理も単純なことである。一方欠点は、キーの範囲が小さくないと使えないことである。例えば、大きいデータでは番号は9桁以上だから10億通りの可能性があり、逆引き表は現実的でない。このため、逆引き表はあまり一般的ではない。

#### （4）ハッシュ表

前述のように大きいデータの番号は9桁だが、数は $1000$ 人そこそこである。このため、番号を適当な関数で $0 \sim 1000$ （現実には余裕を見て $1200$ くらいにとる）に写像できれば、逆引き表が使える。これをハッシュ関数といい、ハッシュ関数を使った逆引き表をハッシュ表という。簡単なハッシュ関数としては、番号を配列のサイズで割り算した余り、というのがある。今、配列のサイズを $1201$ とすると、

$h(n) = n \bmod 1201$ （modは剰余を求める演算子）

マスター[h(番号)] = 内容

とすればよい理屈だが、一つ問題がある。一例として、番号は $850604014$ であり、 $1201$ で割った余りは $746$ だが、他にも余り（ハッシュ値）が $746$ がいるかもしれない。これを衝突（collision：コリジョン）という。衝突があった場合の処理にはいろいろあるが、単純な対処としては、隣の欄を用いていく方法などがある。ハッシュ法は登録・検索とも極めて効率がよく、データ量が増えても検索の手間が変わらないという際立った特徴がある。にも関わらずハッシュ法が実務ではあまり使用されないのは、ハッシュ法を知らない人が多いし、メモリ上のハッシュは簡単だが、ディスクファイル上では手間が掛かる、などの理由が想起される。上記のハッシュ関数は非常に単純だが、文字列（例えば氏名）によるハッシュ関数にはもう少し工夫が必要である。文字列のハッシュとして以下のような関数を用いている。

$h = (\dots((s[1] * 37 + s[2]) * 37 + s[3]) * 37 \dots) s[n]) * 37$

ハッシュ関数は、上記の値を「ててててて」たてて

る。上記は「線形合同法」という乱数生成法と良く似ている。他の著名な乱数生成法の中では「平方探中法」もハッシュ関数として利用できる。また、ハッシュ関数は、元の値からできるだけ重複しない値を作り出すわけだから、データの特徴を示す「電子指紋」としてハッシュ関数と類似の関数を使用されるケースがある。この場合は関数の値域を十分大きくし、関数を工夫することにより、現実のデータではまず重複の起り得ない関数が工夫されている。電子指紋は、データが改竄（かいざん）されていないことを示す場合などに使用されており、電子商取引などに重要である。さらに、ハッシュ関数は元のデータからでたらめな値を作り出すことから、一種の暗号化のことをハッシュと呼ぶ場合があり、ハッシュという語感にはぴったりくるが、これは不正確な用法である。1201は素数である。表のサイズは素数が好ましい。性能は表の利用率によって異なる。利用率8割の場合、もっとも素朴な方法でも平均3回くらいの操作で検索可能である。ハッシュ表はデータが詰まってくると効率が悪くなるので、ある程度詰まってきたら（例：利用率90%）、表のサイズを大きくしてデータを詰め直すこともある。これを再ハッシュ（rehash）という。ハッシュ（hash）は英語で「切り刻む」という意味があり、hashed beefからハヤシライスの語源となった。

【010】検索という操作はプログラミングにおいて非常に頻繁に使われるもので、対象となるデータ量が比較的少ない場合には単純に順番に調べても最近の高速なマシンでは問題無い速度が得られるが、データ量が大きくなってきたり、頻繁に検索する必要がある場合には速度は非常に重要である。検索に関しては非常に多くの文献があり、詳細な説明は、文献を見ていただくのが良い。とりあえずここに実現プログラムを作るのに必要になる事を簡単に示す。

#### 線形サーチ

検索というと真っ先に思い付くのがこの方法で、単純にデータを先頭から順番にアクセスし、探したいものを見つける。この方式のメリットはデータの順序がバラバラでかまわない点と、簡単にすぐに理解できて作ることが出来る点である。あとで出て来るバイナリサーチなどでは検索前にデータを整列しておく必要がある。デメリットは速度が遅い場合が多い点で、データの先頭の方に探したいものがあれば速いが、最悪の場合は全てのデータを見ることになってしまう。したがって、データ量が多ければ多い程速度の問題が大きくなって来る。なお、Cのライブラリに**lsearch()**、**lfind()**という線形サーチの関数がある。

#### バイナリサーチ

プログラマーの間で最も人気のある方式で、先の線形サーチに比べて比較的簡易な準備だけで高速な検索が行える

して、データが昇順にソートされている必要がある。これが実はこの方式のネックになるが、データが比較的固定的で、一度ソートしておいて、あとは検索を繰り返す行なうのみ、といった場合に良い。しかし、検索とデータの変更・追加が両方とも頻繁な場合には、データが変わるたびにソートしなくてはならない為、いくら検索が速くてもソートの時間がかかって総合的には全く速くない、という現象も起こり得る。この点には十分注意する必要がある。ソートは一般的に検索よりも時間がかかる。仕組みとしてはデータがソートされているので、適当に当たりをつけて（真中当たり）比較し、対象より大きければそれより手前の真中当たりで比較する、といった感じで探すので、全データを見る必要はない。はじめにデータをソートする。この時に使用する比較関数と検索で使う比較関数は同じ物を使わないと意味がない。ソートしたデータに対して、見つければそのアドレスが返る。見つからない場合はNULLが返る。見つかった時に配列のいくつめかを表示してる。一般的には構造体の配列などを探すのに使われ、構造体のメンバーで検索して、対象を得る場合には便利である。この検索方式は全データを見ずに結果が得られるので、先の線形検索に比べて高速である。ソートが問題なのですが、ソートにもいろいろなアルゴリズムがありますが、一般的にはここで使ったクイックソートが簡単で高速である。ただ、ソートも万能な物は無く、もとのデータのならびに特徴がある場合にはそれにあったアルゴリズムを使用すべきだ。クイックソートは比較的データの分散に依存せずに平均的に高速にソートする点が便利で良く使われる。

#### ハッシュサーチ

ハッシュサーチでは検索自体は圧倒的に高速である。仕組みとしてはデータをテーブルに格納する際に簡単な式でキーを割り当て、そのキーでダイレクトに飛べる場所にデータを格納しておく。検索の際は同様の式でキーを計算し、ダイレクトにその場所を得る。たとえば、文字列を格納したい場合にはその文字列のコードを全て足したものをキーとする。格納用のテーブルは普通は無制限には取れないので、全て足したものをテーブルの数で割った余りを実際のキーにする。キーが重複する可能性も十分あるので、それに対応する為、格納先を配列形式にしておいたり、そのキーから空いたテーブルを順に探したりする方法が取られている。テーブルに格納したい文字列を引数で渡し、格納先のキーが返る。そのキーを渡すと文字列が返る。文字列を渡し、それがテーブルに存在するかどうかを調べ、存在すればキーを返す。これらを使用すると、文字列を数値として管理が出来るようになり、長さ制限の無い文字列をプログラム中で整数として扱えるようになる。ある文字列に対して必ず唯一のキーが割り当てられるので、そのキーの値で比較なども行える。文字列の文字コードを全て足し、配列のキー

キーにしている。一応、0はエラー判定に使う為に+1した値を使っている。このようにハッシュサーチではキーを計算するだけで格納先に飛ぶので、データの個数がどれだけ増えても検索の時間は変わらない。逆検索および、登録の際ははじめのキーの計算先の配列の個数が増えて来ると徐々に時間がかかるようになるが、それでもはじめのキーの計算で分散性を良くしておけばあまり致命的に偏らないかぎりにはそれほど速度が遅くなることはない。ハッシュサーチでは管理は面倒で、しかも、削除も出来ない、或いはしないほうが良いといった個性的な面もあるが、特に前記したような文字列のコード化などでは圧倒的な強さを見せる。登録した内容を変更しなくて良い場合には最適である。Cのライブラリにも`hsearch()`という関数が用意されているが、1つしかテーブルを持ってない点など不満もある。キーを計算するところでループを使っているが、そこのループの回数を減らしたりして高速化も可能だ。ここでは線形サーチ・バイナリサーチ・ハッシュサーチを取り上げたが、ツリー構造のデータを使った検索など、まだまだいろいろなアルゴリズムがある。万能な物はないので、高速な検索が必要となった場合は、データ構造から十分に検討し、最適な検索を用いる必要がある。いろいろな検索に対応する為に各種の検索専用テーブルを用意したりしたこともありますが、データを更新した際に検索用テーブルも併せて更新する手間や、テーブル自体のメモリー使用量を考えると、後付け方式はあまり効果が無い場合が多い。

【011】次ぎに具体的な攻撃について考えよう。まず、クラッカーによる第1の種類の攻撃として、一般にポートスキャン(`Port Scan`)と言われる種類の攻撃がある。この攻撃は、ネットワークに直接的な損害を及ぼすものではないが、その前段階の攻撃として用いられることが多い。この攻撃では、クラッカーは、自身の管理下にあるホストから、攻撃対象のネットワークに対して、パケット内の宛先IPアドレスや宛先ポート番号を適宜変更しながらIPパケットを繰り返し送信する。そして、それらのIPパケットに対する応答を上記ホストを介して観測する。これにより、攻撃対象のネットワークにおいて、ファイヤウォール等による制限を受けずに外部との通信に利用されているIPアドレスやポート番号を探索する。なお、ここで、前記ポート番号は、TCPあるいはUDP上で動作するアプリケーションソフトウェアのサービス種類(例えば`telnet`、`ftp`、`smtp`、`tfpp`等)を表すもので、IPパケット内のTCPヘッダあるいはUDPヘッダに付与されるデータである。この種の攻撃では、上記のようなIPパケットの送信は、通常、専用のツールソフトウェアを用いて行われ、攻撃対象のネットワークには、宛先IPアドレスやポート番号が互いに異なり、且つ宛先

短時間内に多数、送信される。そこで、本発明では、前記攻撃検知手段は、取得して保持した前記複数のIPパケットのうち、前記ネットワークにその外部から所定時間内に送信されてきた複数のIPパケットであって、少なくともその送信元IPアドレスが互いに同一で且つ宛先IPアドレス又は宛先ポート番号が互いに異なるものが所定数以上あるとき、第1の種類の前記攻撃がなされたことを検知する。これにより、ポートスキャンと言われる第1の種類の攻撃を確実に検知することができる。次に、クラッカーによる第2の種類の攻撃として、一般に`Syn-flood`と称される種類の攻撃がある。この攻撃は、TCPの特性を利用してネットワーク内の特定のホストをダウンさせるものである。すなわち、TCPでは二つのホスト間で通信を行う場合、まず、両ホスト間で論理的なコネクションの開設処理が行われる。このコネクション開設処理では、一方のホストから他方のホストに対して`Syn`用IPパケットを送信する。ここで、該`Syn`用IPパケットは、それを詳しく言えば、上記一方のホストのIPアドレスと他方のホストのIPアドレスとをそれぞれ送信元IPアドレス、宛先IPアドレスとしたIPパケットで、そのパケット内のTCPヘッダの`Syn`ビット及び`Ack`ビットのうちの`Syn`ビットのみを「1」としたものである。そして、コネクション開設処理では、この`Syn`用IPパケットを受けた他方のホストは、前記一方のホストに対して`Syn/Ack`用IPパケットを送信する。ここで、該`Syn/Ack`用IPパケットは、詳しくは、上記他方のホストのIPアドレスと一方のホストのIPアドレスとをそれぞれ送信元IPアドレス、宛先IPアドレスとしたIPパケットで、そのパケット内のTCPヘッダの`Syn`ビット及び`Ack`ビットを共に「1」としたものである。さらに、コネクション開設処理では、この`Syn/Ack`用IPパケットを受けた前記一方のホストは、前記他方のホストに対して`Ack`用IPパケットを送信し、この`Ack`用IPパケットを前記他方のホストが受けることで、両ホスト間の論理的なコネクションの開設がなされる。なお、上記`Ack`用IPパケットは、詳しくは、前記`Syn`用IPパケットと同一の送信元IPアドレス及び宛先IPアドレスを有するIPパケットで、そのパケット内のTCPヘッダの`Syn`ビット及び`Ack`ビットのうちの`Ack`ビットのみを「1」としたものである。

【012】前記`Syn-flood`は、このようなTCPの特性を利用する攻撃である。この攻撃では、クラッカーは、攻撃対象のネットワークの特定のホストに対して、比較的短い時間内に多数の`Syn`用IPパケットを送信する。そして、それらの各`Syn`用IPパケットに対して上記特定ホストから`Syn/Ack`用IPパケットが送信されて来ても、`Ack`用IPパケットとしての特

き、上記特定ホストは、最初に送信されてきたSyn用IPパケットに対するSyn/Ack用IPパケットを送信した後、所定時間（一般に2分）は、その時間内にAck用パケットが送信されてこない限り、そのAck用パケットの受信待ち状態となる。そして、この状態で新たなSyn用パケットが送信されてくる毎に、上記特定ホストは、新たなSyn用パケットに応じたコネクション開設処理を順番に完結すべくその新たなSyn用パケットの情報を通信処理用のバッファ領域に蓄積していく。ところが、バッファ領域の大きさには限界があり、該バッファ領域が満杯になると、前記特定ホストは、TCPの通信処理やTCP上のサービス処理を行うことができなくなる。これにより、特定ホストがダウンすることとなる。この種の攻撃（Syn-flood）では、前述のように、比較的短い時間内に、比較的多くのSyn用IPパケットが攻撃対象のネットワーク内の特定のホスト（特定のIPアドレスを有するホスト）に対して送信されてくる。また、これに応じて、当該特定のホストからネットワークの外部に向かって、比較的短い時間内に、多くのSyn/Ack用IPパケットが送信される。さらに、それらのSyn用IPパケットあるいはSyn/Ack用IPパケットに対応して最終的に前記特定ホストに送信されてくるべきAck用パケットがその特定ホストに送信されてこない。そこで、本発明では、前記攻撃検知手段は、取得して保持した前記複数のIPパケットのうち、前記ネットワークにその外部から所定時間内に送信されてきたTCP（Transmission Control Protocol）に基づく複数のSyn用IPパケットであって、少なくともその宛先IPアドレスが互いに同一であるものが所定数以上あり、且つ、その各Syn用IPパケットと同一の送信元IPアドレス及び宛先IPアドレスを有すると共に前記TCPに基づくAck用IPパケットが前記所定時間内に取得されていないとき、第2の種類の前記攻撃がなされたことを検知する。あるいは、前記攻撃検知手段は、取得して保持した前記複数のIPパケットのうち、前記ネットワークからその外部に所定時間内に送信されたTCP（Transmission Control Protocol）に基づく複数のSyn/Ack用IPパケットであって、少なくともその送信元IPアドレスがそれぞれ互いに同一であるものが所定数以上あり、且つ、前記TCPに基づくAck用IPパケットであって、前記各Syn/Ack用IPパケットの送信元IPアドレス及び宛先IPアドレスとそれぞれ同一の宛先IPアドレス及び送信元IPアドレスを有するものが前記所定時間内に取得されていないとき、第2の種類の前記攻撃がなされたことを検知する。これにより、Syn-floodといわれる第2の種類の攻撃を確実に検知する。次に、クラッカーによる第2の種類の攻

撃がある。この攻撃は、IPパケットの分轄（所謂IPフラグメント）に係る処理の特性を利用してネットワーク内の特定のホストをダウンさせるものである。すなわち、IPパケットは、インターネット上をルータを介して転送される過程で、各ルータのデータ処理容量の関係上、分轄されることがある。また、各ルータにおいてIPパケットが転送される際にエラーが生じることもあり、このような場合には、ルータは、IPパケットの再送信を行う。このため、IPパケットの宛先IPアドレスのホストでは、分轄された一部の同じIPパケットが、複数受信されるということもある。このようなことから、IPに基づく通信では、最終的にIPパケットを受け取るホスト（宛先IPアドレスのホスト）は、受け取ったIPパケットが分轄されたものであるとき、残りの全ての分轄部分のIPパケットを受信するまで、各分割部分のIPパケットを蓄積保持する。そして、全ての分轄部分のIPパケットを受信してから、それらを整理して元のIPパケットのデータを復元する処理を行う。前記Teardropは、このようなIPパケットの分轄に係る処理の特性を利用する攻撃である。この攻撃では、クラッカーは、比較的短い時間内に、多数の同じ分轄部分のIPパケットを攻撃対象のネットワークの特定のホストに送信した上で、残りの分轄部分のIPパケットをその特定ホストに送信する。このような攻撃がなされたとき、上記特定ホストは、最終的に残りの分轄部分のIPパケットを受信したときに、そのIPパケットと、先に送信されてきた多量の分割部分のIPパケットとから元のIPパケットのデータを復元しようとする処理を行うため、その処理に長時間を要するものとなる。このため、該特定ホストは、事実上、ダウンしてしまうこととなる。この種の攻撃（Teardrop）では、前述の如く、比較的短い時間内に、多数の同じ分轄部分のIPパケットがネットワーク内の特定のホストに送信されてくる。そこで、本発明では、前記攻撃検知手段は、取得して保持した前記複数のIPパケットのうち、前記ネットワークにその外部から所定時間内に送信されてきた複数の分割されたIPパケットであって、同一の分割部分が所定数以上あるとき、第3の種類の前記攻撃がなされていることを検知する。これにより、Teardropといわれる第3の種類の攻撃を確実に検知することができる。次に、クラッカーによる第4の種類の攻撃として、一般にLandと称される種類の攻撃がある。この攻撃は、送信元IPアドレス及び宛先IPアドレスが同一であるような、正規にはあり得ないIPパケットを、攻撃対象のネットワークの特定のホストに送信する攻撃である。このようなIPパケットを送信された特定ホストは、そのIPパケットの処理に手間取ることが多く、ダウンしてしまうことがしばしばある。

【012】この種の攻撃では、上記の如く、送信元IP



トが、ネットワーク内の特定のホストに送信される。しかも、一般には、そのようなIPパケットが比較的短い時間内に、複数、上記特定ホストに送信される。そこで、本発明では、前記攻撃検知手段は、取得して保持した前記複数のIPパケットのうち、前記ネットワークにその外部から所定時間内に送信されてきた複数のIPパケットであって、その送信元IPアドレスが宛先IPアドレスと同一のアドレスとなっているものが所定数以上あるとき、第4の種類の前記攻撃がなされていることを検知する。これにより、Landとわれる第4の種類の攻撃を確実に検知することができる。なお、前述したSyn-flood、Teardrop、Landといわれる攻撃は、一般に、DoS (Denial of Service) といわれる種類の攻撃に属するものである。そして、このDoSには、Syn-flood、Teardrop、Landのほか、例えばSmurfといわれる種類の攻撃や、Floodieといわれる種類の攻撃等もある。本明細書では、DoSに属する種類の攻撃として、代表的にSyn-flood、Teardrop、Landを挙げたが、SmurfやFloodie等の攻撃を検知するようにすることも可能である。前述のようにクラッカーによる攻撃を検知する攻撃検知手段を備えた本発明では、前記処理手段が行う処理は、例えば前記攻撃が検知された旨を表す報知出力を発生する処理である。この報知出力の発生により、ネットワーク管理者やあるいは外部の専門技術者等が、検知された攻撃を排除するための処置を施すことが可能となる。あるいは、前記処理手段が行う処理は、前記攻撃検知手段が検知した前記攻撃に係る特定の送信元IPアドレス及び／又は宛先IPアドレスを有するIPパケットの前記ネットワークへの進入を、前記攻撃を検知してから所定時間阻止する処理である。

【014】高速検索アルゴリズムとしては、タイムキュー付きのハッシュ法を利用している。その具体的な特徴は以下の実施例に示した。一方、コーディング方式としては、LINUXのkernel-codingを採用した。従って、OSの機能を利用せず、直接インターフェースとやりとりをするプログラムを作製した。これは開発にかなりの時間を要する事になったが、できたソフトの速度は大変高いものとなった。一般に、インターフェースとのやりとりはOSの主要な仕事であり、OSの利用者はそれをインターフェースの付いたサブルーチンとして提供を受ける事ができる。そうすれば、通常のUNIX (登録商標) ユーザーには公開されていない、例えばメモリー管理情報にもアクセスする必要がなくなるのだが、ここでは、速度を重視したため、直接駆動するコードを作製した。OSはいろいろなハードウェア環境全体を効率よく管理する目的の為には、有望だが、単一の機能に特化したパフォーマンス向上のためには

での開発は行わないため、比較的短時間で開発が済む代わりに、性能的には低速になっていた。今回、世界で始めて開発された部分である。単なる検出だけなら、ここまでの高速性は不要である、というのは、機械から見ると遅くとも、人間にとって充分早い程度の時間 (例えば1~2秒) の内に、記録を残せば充分であったからである。今回、我々は、パケットが通過するまでの時間 (1/1000秒程度) に、判断して遮断する必要があったため、高速のアルゴリズムおよび、高速のプログラム実行方式を用意せざるを得なかったのである。そして、高速で処理の実をフルに生かした製品が出来上がっている。他にも、ハッシュ値の表が時間とともに変化する機能が存在している。古いパケットの情報は自動的に消滅するのである。インターネットの攻撃、それも特にDoSと言われる攻撃に対抗するには、一定の時間内にある程度の数の攻撃パケットに対応できる能力が必要になる。これの作り込みに、タイムキューの果たす役割は重要になっている。また、各パケットの間が、あまりに長時間の場合はDoSとしての攻撃の効果は薄れるため、消去の設定が必要となる。

#### 【015】

【発明の実施の形態】 本発明の一実施形態を図1を参照して説明する。図1は本実施形態のシステム構成図である。図1において、1はネットワークとしてのLANである。このLAN1は、例えばイーサネット (登録商標) (Ethernet (登録商標)) を用いて構築されたものであり、図示を省略する複数のホスト (コンピュータ) がイーサネット・ケーブルやハブ等を介して接続されている。各ホストには、それをイーサネット・ケーブルに接続するイーサネット・カードや、TCP/IPの処理を行うためのソフトウェア、TCP/IP上で機能する各種アプリケーションソフトウェア (例えば、telnet、ftp、smtp等) が実装され、IPに基づく通信を可能としている。なお、LAN1は、イーサネット上で構築されたものに限らず、トークンリング等、他の形態で構築されたものであってもよい。本実施形態のシステムでは、LAN1の入りに、パケットフィルタとしてのファイヤウォールの機能をもたせたコンピュータ2 (以下、このコンピュータ2を単にファイヤウォール2と称する) が設けられている。そして、LAN1はファイヤウォール2を介してインターネット3に接続されている。ファイヤウォール2は、どのような種類のIPパケットのLAN1への進入を禁止するかを規定するデータが書き込まれるファイル (以下、フィルタ設定ファイルという) を有している。そして、ファイヤウォール2は、このフィルタ設定ファイルで、LAN1への進入が禁止された種類のIPパケットがインターネット3側から送信されてきたときに、そのIPパケットを遮断してLAN1への進入を阻止する。また、フ

ないIPパケットが送信されてきたときには、それをLAN1に転送する。ファイヤウォール2とインターネット3との間には、ハブ4が介装され、このハブ4に攻撃検知手段の機能をもたせたセンサ5が接続されている。また、このセンサ5には、前記ファイヤウォール2を制御する処理手段の機能を有するディレクタ6が接続されている。これらのセンサ5及びディレクタ6はそれぞれコンピュータにより構成されたものである。前記センサ5は例えばUNIXマシンにより構成され、イーサネットカード7を介して前記ハブ4に接続されている。この場合、センサ5には、tcpdumpといわれるソフトウェアが実装されている。このtcpdumpによって、ハブ4を通る全てのIPパケットをイーサネットカード7を介して取得する（ヒアリングする）ことができる。このような動作は、プロミス・キャスト・モード（promise cast mode）といわれることが多い。そして、センサ5は、取得した各IPパケットをその取得時点の時刻データと共に図示しないハードディスクに記憶保持するようにしている。なお、ハードディスクに記憶保持したIPパケットの総量が所定の許容量に達したときには、センサ5は、最も古いIPパケットを消去し、新たに取得されたIPパケットをハードディスクに記憶保持する。また、センサ5は、IPアドレスを持たず、ARP（Address Resolution Protocol）や、RARP（Reverse Address Resolution Protocol）のパケット等、応答を促すパケットが送信されてきても、それに対する応答をしないようにソフトウェア的に設定されている。つまり、センサ5はIPパケットの受信（取り込み）のみを行うことができるものとされている。さらに、センサ5には、前述した第1～第6の種類の攻撃を検知するためのソフトウェア（以下、攻撃検知アルゴリズム）が実装されている。なお、この攻撃検知アルゴリズムは、ディレクタ6に実装しておき、該ディレクタ6とのデータ授受を行いつつ該攻撃検知アルゴリズムの処理をセンサ5に行わせるようにしてもよい。前記ディレクタ6には、前記ファイヤウォール2を制御するソフトウェア（以下、フィルタ制御アルゴリズムという）が実装されている。この場合、フィルタ制御アルゴリズムは、センサ5により検知される攻撃に応じて、前記フィルタ設定ファイルのデータを適宜書き換えることで、前記ファイヤウォール2を制御するものである。

【016】次に、かかる本実施形態の作動を説明する。前記センサ5は、取得されるIPパケットを前述の如くハードディスクに記憶保持しつつ、所定のサイクルタイム毎に次のような処理を行う。すなわち、センサ5は、ハードディスクから所定の時間間隔分の複数のIPパケットを、送信元IPアドレス及び宛先IPアドレスの値

のうち、同一の送信元IPアドレスを有するものをひとまとめにすると共に、同一の宛先IPアドレスを有するものをひとまとめにして、メモリに取り込む（以下の説明では、このようにひとまとめにされたIPパケットの組をIPパケット群という）。そして、このメモリに取り込んだ複数のIPパケットに対して、後述する攻撃検知の処理を行った上で、それらのIPパケットをメモリから消去する。この場合、各サイクルタイムにおいて、メモリに取り込むIPパケットは、前回のサイクルタイムでメモリに取り込んだIPパケットのうちの最も古いIPパケットの取得時刻から所定時間を経過した時刻以後に取得されたものである。各サイクルタイムにおけるセンサ5による攻撃検知の処理は、攻撃検知アルゴリズムに従って次のように行われる。センサ5は、まず、前記第1～第6の種類の攻撃のうち、例えば、第1の種類の攻撃、すなわちポートスキャンを検知する処理を行う。この処理では、センサ5は、メモリに前述のように取り込んだIPパケットのうち、送信元IPアドレスが同一で、且つ該送信元IPアドレスがLAN1の外部のものである各IPパケット群に対し、その各IPパケット群に含まれるIPパケットが有する全ての宛先IPアドレスの値（これはLAN1に属するIPアドレスの値である）を抽出する。そして、上記の各IPパケット群で抽出した宛先IPアドレスの各値に対し、そのIPパケット群（同一の送信元IPアドレスのIPパケット群）から、該宛先IPアドレスの値と同一の宛先IPアドレスを有し、且つTCPヘッダあるいはUDPヘッダ内の宛先ポート番号が互いに異なり、且つ、連続した所定時間内（例えば30秒内）に取得されたIPパケットの個数をカウントする。このとき、このカウント数が所定数（例えば20個）に達した場合には、センサ5は、ポートスキャンの攻撃がなされていることを検知する。そして、そのことを示すデータと、この攻撃が検知されたIPパケット群の送信元IPアドレスの値データとを（以下、これらのデータを第1種攻撃検知データという）前記ディレクタ6に与える。このような処理が送信元IPアドレスが同一で、且つ該送信元IPアドレスがLAN1に属さない全てのIPパケット群に対し順次行われる。なお、本実施形態におけるポートスキャンの検知では、ポート番号が互いに異なるIPパケットの個数をカウントするようにしたが、次のような処理によりポートスキャンを検知するようにしてもよい。すなわち、送信元IPアドレスが同一で、且つ、該送信元IPアドレスがLAN1外部のものである各IPパケット群に対し、その各IPパケット群に含まれるIPパケットが有する全ての宛先ポート番号の値を抽出する。さらに、その抽出した宛先ポート番号の各値に対し、該宛先ポート番号を抽出したIPパケット群から、該宛先ポート番号の値

たIPパケットの個数をカウントする。そして、そのカウント数が所定数に達した場合にポートスキャンが行われていることを検知する。一方、センサ5から前述のような第1種攻撃検知データを与えられた前記ディレクタ6は、該第1種攻撃検知データに含まれる送信元IPアドレスと同一の送信元IPアドレスを有するIPパケットがLAN1に進入するのを現在から所定時間（例えば5分間）阻止するように前記ファイヤウォール2のフィルタ設定ファイルを書き換える。このとき、ファイヤウォール2は、上記送信元IPアドレスを有するIPパケットがインターネット3から送信されてくると、そのIPパケットを廃棄し、LAN1への進入を阻止する。これにより、ポートスキャンの攻撃からLAN1が保護される。なお、ディレクタ6は、上記所定時間（5分間）が経過するまでの間に、先に与えられた第1種攻撃検知データと同一の第1種攻撃検知データがセンサ5から再度与えられれば、その時点から上記所定時間（5分間）、該第1種攻撃検知データの送信元IPアドレスからのIPパケットのLAN1への進入を阻止するようにファイヤウォール2を制御する。従って、ポートスキャンの攻撃が続いている限り、その送信元IPアドレスからのIPパケットは、LAN1に進入することはできない。そして、ディレクタ6は、上記所定時間（5分間）が経過するまでに、前記第1種攻撃検知データが与えられなかった場合には、その第1種攻撃検知データの送信元IPアドレスからのIPパケットのLAN1への進入の阻止を解除する。前述のようにポートスキャンの攻撃の検知処理を行ったセンサ5は、次に、第2の種類の攻撃（Syn-flood）の検知処理を行う。

【017】この処理では、センサ5は、宛先IPアドレスが同一であるIPパケット群のうち、LAN1に属する宛先IPアドレスの各IPパケット群に対し、該IPパケット群に含まれるSyn用IPパケットをその取得時刻順に順次抽出する。そして、抽出した各Syn用IPパケットの取得時刻から所定時間（例えば2秒間）内に取得されたSyn用IPパケットが、同じ宛先IPアドレスのIPパケット群内に存在するか否かを調べる。そして、そのようなSyn用IPパケットが存在する場合には、先に抽出したSyn用IPパケットを含めてそれらのSyn用IPパケットの個数をカウントする。さらに、そのカウントしたそれぞれのSyn用IPパケットに対して、それぞれに対応するAck用IPパケット

（詳しくは該Syn用IPパケットと同一の送信元IPアドレスを有し、且つ、該Syn用IPパケットのTCPヘッダ中のシーケンス番号の次のシーケンス番号を有するAck用IPパケット）であって、且つ該Syn用IPパケットの取得時刻から上記所定時間（2秒間）内に取得されたものが、同じ宛先IPアドレスのIPパ

度、上記のカウント数を「1」づつ減少させる。そして、最終的に、対応するAck用IPパケットの存在を調べ終わったときに上記のカウント数が所定数（例えば16個）以上である場合には、Syn-floodの攻撃がなされていることを検知し、そのことを示すデータと、この攻撃が検知されたSyn用IPパケットの送信元IPアドレスの値データ及び宛先IPアドレスの値データとを（以下、これらのデータを第2種攻撃検知データという）前記ディレクタ6に与える。このような処理が宛先IPアドレスが同一で、且つ該宛先IPアドレスがLAN1に属する全てのIPパケット群に対して順次行われる。なお、本実施形態では、Syn用IPパケットの個数に基づいてSyn-floodを検知したが、次のような処理によりSyn-floodを検知するようにしてもよい。すなわち、送信元IPアドレスが同一で且つ、該送信元IPアドレスがLAN1に属する各IPパケット群に対し、該IPパケット群に含まれるSyn/Ack用IPパケットをその取得時刻順に順次抽出する。そして、抽出した各Syn/Ack用IPパケットの取得時刻から所定時間（例えば2秒間）内に取得されたSyn/Ack用IPパケットが、同じ送信元IPアドレスのIPパケット群内に存在するか否かを調べる。このとき、そのようなSyn/Ack用IPパケットが存在する場合には、先に抽出したSyn/Ack用IPパケットを含めてそれらのSyn/Ack用IPパケットの個数をカウントする。さらに、そのカウントしたそれぞれのSyn/Ack用IPパケットに対して、該Syn/Ack用IPパケットの送信元IPアドレスと同一の宛先IPアドレスのIPパケット群を調べる。このとき、該Syn/Ack用IPパケットに対応するAck用IPパケット（詳しくは該Syn/Ack用IPパケットの送信元IPアドレスと同一の宛先IPアドレスを有し、且つ、該Syn/Ack用IPパケットのTCPヘッダ中のシーケンス番号の次のAck番号を有するAck用IPパケット）であって、且つ該Syn/Ack用IPパケットの取得時刻から上記所定時間（2秒間）内に取得されたものが、当該IPパケット群内に存在するか否かを調べる。そして、そのようなAck用IPパケットが存在する場合には、その都度、上記のカウント数を「1」づつ減少させる。そして、最終的に、対応するAck用IPパケットの存在を調べ終わったときに上記のカウント数が所定数（例えば16個）以上である場合には、Syn-floodの攻撃がなされていることを検知する。

【018】なお、この場合にセンサ5からディレクタ6に与えるデータは、Syn-floodの攻撃を検知したことを示すデータと、上記Syn/Ack用IPパケットの送信元IPアドレスの値データ及び宛先IPアドレスの値データである。この場合、Syn/Ack用IP

Pアドレスの値データは、それぞれ、先に説明した前記第2種攻撃検知データにおけるSyn用IPパケット宛先IPアドレスの値データ、送信元IPアドレスの値データに相当するものである。一方、センサ5から前述のような第2種攻撃検知データを与えられた前記ディレクタ6は、該第2種攻撃検知データに含まれる送信元IPアドレスと同一の送信元IPアドレスを有するIPパケットがLAN1に進入するのを現在から所定時間（例えば2分間）阻止するように前記ファイヤウォール2のフィルタ設定ファイルを書き換える。同時に、ディレクタ6は、第2種攻撃検知データに含まれる宛先IPアドレスと同一の宛先IPアドレスを有するIPパケットがLAN1に進入するのを現在から所定時間（例えば2秒間）阻止するようにファイヤウォール2のフィルタ設定ファイルを書き換える。このとき、ファイヤウォール2は、上記送信元IPアドレスを有するIPパケット、あるいは上記宛先IPアドレスを有するIPパケットがインターネット3から送信されてくると、そのIPパケットを廃棄し、LAN1への進入を阻止する。これにより、Syn-floodの攻撃からLAN1が保護されると共に、この攻撃の対象とされていたIPアドレスのホストがダウンせずに正常状態に復帰することができる。なお、ポートスキャンの検知時の場合と同様、ディレクタ6は、第2種攻撃検知データにおける送信元IPアドレスを有するIPパケットの排除に係る上記所定時間（2分間）が経過するまでの間に、先に与えられた第2種攻撃検知データと同一の第2種攻撃検知データがセンサ5から再度与えられれば、その時点から上記所定時間（2分間）、該第2種攻撃検知データの送信元IPアドレスからのIPパケットのLAN1への進入を阻止するようにファイヤウォール2を制御する。このことは、第2種攻撃検知データにおける宛先IPアドレスを有するIPパケットの排除についても同様である。従って、Syn-floodの攻撃が続いている限り、その攻撃に係る送信元IPアドレスからのIPパケット、あるいはその攻撃に係る宛先IPアドレスへのIPパケットは、LAN1に進入することはできない。そして、ディレクタ6は、第2種攻撃検知データにおける送信元IPアドレスを有するIPパケットの排除と、第2種攻撃検知データにおける宛先IPアドレスを有するIPパケットの排除とのいずれについても、それぞれに対応する上記所定時間（2分間、2秒間）が経過するまでに、前記第2種攻撃検知データが与えられなかった場合には、その第2種攻撃検知データの送信元IPアドレスを有するIPパケット、あるいは、第2種攻撃検知データの宛先IPアドレスを有するIPパケットのLAN1への進入の阻止を解除する。前述のようにSyn-floodの攻撃の検知処理を行ったセンサ5は、次に、第3の種類の攻撃（Teardrop）の検知処理を行う。この検知

Pパケット群のうち、LAN1に属する宛先IPアドレスの各IPパケット群に対し、該IPパケット群に含まれる分轄されたIPパケット（以下、単に、分轄パケットという）を順次抽出する。この場合、IPでは、分轄パケットは、そのIPヘッダ中の特定のフラグが「1」となっているか、もしくは、フラグメントオフセットといわれるデータが「0」より大きな値となっている。これにより、分轄パケットを見出すことができる。そして、センサ5は、抽出した各分轄パケットの取得時刻から所定時間（例えば5分間）内に取得され、且つ、該分轄パケットとIPヘッダ中のIP識別番号及びフラグメントオフセットの値がそれぞれ同一であるもの（抽出した分轄パケットと同一の分轄パケット）が、該分轄パケットと同じIPパケット群内にあるかを調べる。このとき、そのような分轄パケットがある場合には、先に抽出した分轄パケットを含めてそれらの分轄パケットの個数をカウントする。そして、このカウント数が所定数（例えば80個）以上である場合には、Teardropの攻撃がなされていることを検知し、そのことを示すデータと、この攻撃が検知された分轄パケットの送信元IPアドレスの値データ及び宛先IPアドレスの値データとを（以下、これらのデータを第3種攻撃検知データという）前記ディレクタ6に与える。このような処理が宛先IPアドレスが同一で、且つ該宛先IPアドレスがLAN1に属する全てのIPパケット群に対して順次行われる。一方、センサ5から前述のような第3種攻撃検知データを与えられた前記ディレクタ6は、前記Syn-floodが検知された場合と全く同じやり方で、ファイヤウォール制御する。すなわち、第3種攻撃検知データに含まれる送信元IPアドレスと同一の送信元IPアドレスを有するIPパケットがLAN1に進入するのを現在から所定時間（2分間）阻止するように前記ファイヤウォール2のフィルタ設定ファイルを書き換える。同時に、第3種攻撃検知データに含まれる宛先IPアドレスと同一の宛先IPアドレスを有するIPパケットがLAN1に進入するのを現在から所定時間（2秒間）阻止するようにファイヤウォール2のフィルタ設定ファイルを書き換える。これにより、Teardropの攻撃からLAN1が保護されると共に、この攻撃の対象とされていたIPアドレスのホストがダウンせずに正常状態に復帰することができる。

【019】上記のようにTeardropの攻撃の検知処理を行ったセンサ5は、次に、第4の種類の攻撃（Land）の検知処理を行う。この処理では、センサ5は、宛先IPアドレスが同一であるIPパケット群のうち、LAN1に属する宛先IPアドレスの各IPパケット群から、該IPパケット群の宛先IPアドレスと同じ値の送信元IPアドレスを有するIPパケットを抽出する。次に、その抽出したIPパケットと同一の宛先IP

同じ送信元IPアドレスを有し、且つ該IPパケットの取得時刻から所定時間（例えば2分間）内に取得されたIPパケットが存在するか否かを調べる。そして、そのようなIPパケットが存在する場合には、先に抽出したIPパケットを含めてそれらのIPパケットの該IPパケットの個数をカウントする。このとき、該カウント数が所定数（例えば6個）以上である場合には、Landの攻撃がなされていることを検知し、そのことを示すデータと、この攻撃が検知されたIPパケットの送信元IPアドレスの値データを（以下、これらのデータを第4種攻撃検知データという）前記ディレクタ6に与える。このような処理が宛先IPアドレスが同一で、且つ該宛先IPアドレスがLAN1に属する全てのIPパケット群に対して順次行われる。一方、センサ5から前述のような第4種攻撃検知データを与えられた前記ディレクタ6は、第4種攻撃検知データに含まれる送信元IPアドレスと同一の送信元IPアドレスを有し、且つ、該送信元IPアドレスと同一の宛先IPアドレスを有するIPパケットがLAN1に進入するのを現在から所定時間（例えば3分間）阻止するように前記ファイヤウォール2のフィルタ設定ファイルを書き換える。このとき、ファイヤウォール2は、上記送信元IPアドレス及び宛先IPアドレスを有するIPパケットがインターネット3から送信されてくると、そのIPパケットを廃棄し、LAN1への進入を阻止する。これにより、Landの攻撃からLAN1が保護される。この場合、ポートスキャンの検知時の場合と同様、ディレクタ6は、第4種攻撃検知データにおける送信元IPアドレスと同一の送信元IPアドレス及び宛先IPアドレスを有するIPパケットの排除に係る上記所定時間（6分間）が経過するまでの間に、先に与えられた第4種攻撃検知データと同一の第4種攻撃検知データがセンサ5から再度与えられれば、その時点から上記所定時間（6分間）、該第4種攻撃検知データの送信元IPアドレス及び宛先IPアドレスを有するIPパケットのLAN1への進入を阻止するようにファイヤウォール2を制御する。従って、Landの攻撃が続いている限り、その攻撃に係る送信元IPアドレス及び宛先IPアドレスを有するIPパケットは、LAN1に進入することはできない。そして、ディレクタ6は、上記所定時間（6分間）が経過するまでに、前記第4種攻撃検知データが与えられなかった場合には、その第4種攻撃検知データの送信元IPアドレスと同一の送信元IPアドレス及び宛先IPアドレスを有するIPパケットのLAN1への進入の阻止を解除する。なお、本実施形態では、第4種攻撃検知データとして、Landの攻撃に係るIPパケットの送信元IPアドレスの値データをディレクタ6に与えるようにしたが、Landの攻撃に係るIPパケットの送信元IPアドレス、宛先IPアドレスは同じ値である。従って、その

ドレスの値をディレクタ6に与えてもよいことはもちろんである。前述のように、Landの攻撃の検知処理を行ったセンサ5は、次に第5の種類の攻撃（パスワードの獲得）を検知する処理を行う。この処理では、センサ5は、宛先IPアドレスが同一であるIPパケット群のうち、LAN1に属する宛先IPアドレスの各IPパケット群に対し、LAN1のホストのユーザ名データ及びパスワードデータを含むIPパケットを抽出する。それらの抽出したIPパケットの中から、ユーザ名データが同一で、且つ、パスワードデータが互いに異なり、且つ、連続した所定時間（例えば2分間）内に取得されたIPパケットの個数をカウントする。このとき、このカウント数が所定数（例えば20個）以上であれば、クラッカーがパスワードを獲得するための第5の種類の攻撃がなされていることを検知し、そのことを示すデータと、この攻撃が検知されたIPパケットの送信元IPアドレスの値データ及び宛先IPアドレスの値データとを（以下、これらのデータを第5種攻撃検知データという）前記ディレクタ6に与える。このような処理が宛先IPアドレスが同一で、且つ該宛先IPアドレスがLAN1に属する全てのIPパケット群に対して順次行われる。一方、センサ5から前述のような第5種攻撃検知データを与えられた前記ディレクタ6は、該第5種攻撃検知データの送信元IPアドレス及び宛先IPアドレスとそれぞれ同一の送信元IPアドレス及び宛先IPアドレスを有するIPパケットがLAN1に進入するのを現在から所定時間（例えば1時間）阻止するように前記ファイヤウォール2のフィルタ設定ファイルを書き換える。このとき、ファイヤウォール2は、上記送信元IPアドレス及びIPアドレスを有するIPパケットがインターネット3から送信されてくると、そのIPパケットを廃棄し、LAN1への進入を阻止する。これにより、パスワードの獲得を狙った第5の種類の攻撃からLAN1が保護される。

【020】なお、ポートスキャンの検知時の場合と同様、ディレクタ6は、第5種攻撃検知データにおける送信元IPアドレス及び宛先IPアドレスを有するIPパケットの排除に係る上記所定時間（1時間）が経過するまでの間に、先に与えられた第5種攻撃検知データと同一の第5種攻撃検知データがセンサ5から再度与えられれば、その時点から上記所定時間（1時間）、該第5種攻撃検知データの送信元IPアドレス及び宛先IPアドレスを有するIPパケットのLAN1への進入を阻止するようにファイヤウォール2を制御する。従って、第5の種類の攻撃が続いている限り、その攻撃に係る送信元IPアドレス及び宛先IPアドレスを有するIPパケットは、LAN1に進入することはできない。そして、ディレクタ6は、上記所定時間（1時間）が経過するまでに、前記第5種攻撃検知データが与えられた場合

及び宛先IPアドレスを有するIPパケットのLAN1への進入の阻止を解除する。前述のように、第5の種類の攻撃の検知処理を行ったセンサ5は、次に第6の種類の攻撃（セキュリティホールの攻撃）を検知する処理を行う。この処理では、センサ5は、宛先IPアドレスが同一であるIPパケット群のうち、LAN1に属する宛先IPアドレスの各IPパケット群に対し、例えばプリンタの論理名である「Ipr」を有し、且つ、データサイズが128文字以上であるIPパケットを検索する。そして、そのようなIPパケットが見つかった場合には、LAN1のホストのスルーホールを攻撃する第6の種類の攻撃がなされていることを検知し、そのことを示すデータと、この攻撃が検知されたIPパケットの送信元IPアドレスの値データ及び宛先IPアドレスの値データとを（以下、これらのデータを第6種攻撃検知データという）前記ディレクタ6に与える。一方、センサ5から前述のような第6種攻撃検知データを与えられた前記ディレクタ6は、該第6種攻撃検知データの送信元IPアドレス及び宛先IPアドレスとそれぞれ同一の送信元IPアドレス及び宛先IPアドレスを有するIPパケットがLAN1に進入するのを現在から所定時間（例えば4時間）阻止するように前記ファイアウォール2のフィルタ設定ファイルを書き換える。このとき、ファイアウォール2は、上記送信元IPアドレス及びIPアドレスを有するIPパケットがインターネット3から送信されてくると、そのIPパケットを廃棄し、LAN1への進入を阻止する。これにより、LAN1のホストのスルーホールを攻撃する第6の種類の攻撃からLAN1が保護される。なお、ポートスキャンの検知時の場合と同様、ディレクタ6は、第6種攻撃検知データにおける送信元IPアドレス及び宛先IPアドレスを有するIPパケットの排除に係る上記所定時間（4時間）が経過するまでの間に、先に与えられた第6種攻撃検知データと同一の第5種攻撃検知データがセンサ5から再度与えられれば、その時点から上記所定時間（4時間）、該第6種攻撃検知データの送信元IPアドレス及び宛先IPアドレスを有するIPパケットのLAN1への進入を阻止するようにファイアウォール2を制御する。従って、第6の種類の攻撃が続いている限り、その攻撃に係る送信元IPアドレス及び宛先IPアドレスを有するIPパケットは、LAN1に進入することはできない。そして、ディレクタ6は、上記所定時間（4時間）が経過するまでに、前記第6種攻撃検知データが与えられなかった場合には、その第5種攻撃検知データの送信元IPアドレス及び宛先IPアドレスを有するIPパケットIPパケットのLAN1への進入の阻止を解除する。以上説明したようにして、本実施形態のシステムによれば、センサ5や、ディレクタ6を導入するだけで、クラッカーによるLAN1への各種の攻撃をリアルタイムで検知し、かつ

検知された攻撃からLAN1を保護する適正な処置を自動的に迅速に施すことができる。このため、ネットワーク管理者等は、クラッカーによる攻撃を考慮してLAN1を構築したり、頻繁にログファイルを参照したりする労力が大幅に削減され、ひいては、LAN1の維持管理のコストを低減することができる。また、クラッカーによる各種攻撃をリアルタイムで検知できることから、攻撃が検知されない状況では、LAN1と外部との通信を格別に制限する必要性が少なくなる。このため、通常時は、LAN1の通信の自由度を高めることができ、インターネット3上の情報資源を有効に活用することができる。なお、以上説明した実施形態では、LAN1の入り口にファイアウォール3を設けておき、クラッカーによる攻撃が検知されてとき、該ファイアウォール3を制御することで、検知された攻撃を自動的に排除する処置を行った。但し、クラッカーによる攻撃が検知されたときに、単に、その旨の報知をネットワーク管理者や、専門の警備管理者等に行うようにしてもよい。この場合には、例えば前記ディレクタ6あるいはセンサ5を公衆回線や専用回線を介してネットワーク管理者や、警備管理者等のホストに接続しておく。そして、攻撃が検知された場合に、前述した第1乃至第6種攻撃検知データのような情報をネットワーク管理者や警備管理者等のホストにディレクタ6あるいはセンサ5から送信する。このようにしたときには、検知された攻撃からLAN1を保護するための具体的な処置は、ネットワーク管理者等が直接的に行うこととなる。しかるに、この場合であっても、ネットワーク管理者等は、上記の報知を受けたときに処置を施せばよく、しかも攻撃の種類は検知されるので、攻撃に対する処置を比較的容易に施すことができる。また、前記実施形態では、第1乃至第6の種類の攻撃を順番に検知するものを示したが、それらの攻撃の検知処理を並列的に行うようにすることも可能である。また、前記実施形態では、前述したDoS（Denial of Service）に属する攻撃のうち、Syn-flood、Teardrop、Landを検知するものを示した。但し、この他にも、DDoS（distributed Denial of Service）SmurfやFloodieといわれるような攻撃を検知するようにすることも可能である。

#### 【021】産業上の利用可能性

以上のように、本発明のクラッカー監視システムは、企業や官庁等の組織に構築されたLAN等のネットワークをクラッカーによる攻撃から簡易に保護し、また、その保護を通信の自由度を必要以上に損なうことなく行うことができるシステムとして有用である。

#### 【図面の簡単な説明】

【図1】 図1は、本発明のクラッカー監視システムの一実施形態のシミュレーション構成図である。

【図1】

FIG. 1

